Aufgabe: Mit Hilfe eines Arduino-Mikrocontrollers soll ein RFID-TAG ausgelesen werden. Sofern es sich um den richtigen TAG handelt, soll eine Leuchtdiode für 5 Sekunden leuchten.

Material:

- Arduino-Board
- RFID-Reader
- Mindestens einen RFID-TAG
- Eine LED
- Breadboard
- Steckkabel
- Ein 100 Ohm Widerstand

In dieser Anleitung lernst du:

- RFID-Reader verkabeln
- RFID-Codes verarbeiten
- Codes im HEX-Format umwandeln

Der RFID („radio-frequency identification“) Reader wird verwendet, um von RFID Sendern (auch „RFID Tags“ genannt) per Funk einen bestimmten Code auszulesen. Jeder Sender hat dabei nur einen einmaligen ganz individuellen Code. Damit lassen sich mit dem Arduino Schließanlagen oder ähnliche Projekte verwirklichen, bei denen sich eine Person mit einem Sender identifizieren soll.
RFID-TAGs können verschiedene Formen haben, wie z.B. Schlüsselanhänger oder Karten im Kreditkartenformat.



Auf dem Bild sieht man Links oben und unten rechts zwei RFID-TAGs und in der Mitte den RFID-Empfänger RFID-RC522, mit bereits angelöteter 90° Stiftleiste. Es gibt auch Versionen bei denen die Stiftleiste selbst an den RFID-Empfänger gelötet werden muss.

Wie funktioniert das ganze? Ein RFID-Empfänger enthält eine kleine Kupferspule, die ein magnetisches Feld erzeugt. Ein RFID-Sender enthält ebenfalls eine Kupferspule, die das magnetische Feld aufgreift und in dem Sender eine elektrische Spannung erzeugt. Diese Spannung wird dann verwendet um einen kleinen elektronischen Chip dazu zu bringen, per Funk einen elektrischen Code auszusenden. Dieser Code wird dann direkt vom Sender empfangen und so verarbeitet, dass der Arduino-Mikrocontroller den empfangenen Code weiterverarbeiten kann.

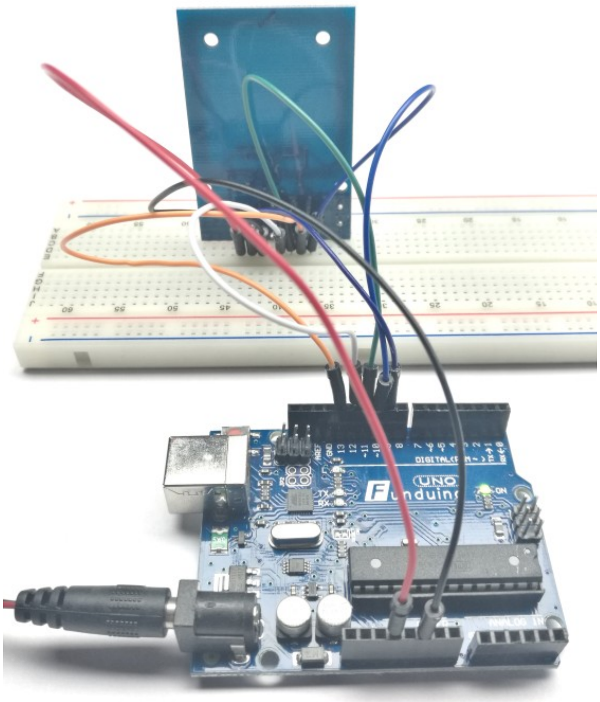
Es ist auch möglich, RFID-TAGs zu mit einem Code zu beschreiben. Dies wird aufgrund der Komplexität in dieser Anleitung nicht behandelt. Weiterführende Tutorials zu dem Themengebiet gibt es im Internet.

Einen RFID-TAG mit Arduino auslesen und die Daten verarbeiten

Verkabelung des Arduino-Boards mit dem RFD-Empfänger:

Board:	Arduino Uno	Arduino Mega	MFRC522-READER
Pin:	10	53	SDA
Pin:	13	52	SCK
Pin:	11	51	MOSI
Pin:	12	50	MISO
Pin:	unbelegt	unbelegt	IRQ
Pin:	GND	GND	GND
Pin:	9	5	RST
Pin:	3,3V	3,3V	3,3V

Aufbau mit um 90° gebogene Kontaktstiften:





Progammierung

Beim Auslesen und verarbeiten der Daten eines RFID-Empfängers wären wie auch bei anderen komplexen Aufgaben sehr viele Zeilen Quellcode erforderlich. Daher bedienen wir uns einer vorgefertigten Library. Die „ MFRC522“ Library von GithubCommunity kann über den Bibliothekenverwalter mit dem Suchbegriff „RFID“ gefunden und installiert werden. Eine Anleitung zur Installation einer Programmbibliothek über den Bibliotheksverwalter befindet sich im Theorieteil zur Arduino Software.

Vorbereitung – der erste Sketch mit dem RFID-READER

Zunächst werden wir die UID („Unique Identification Number“), also die individuelle Bezeichnung eines RFID-TAGs auslesen. Dazu verwenden wir den folgenden Sketch (Achtung, der Sketch funktioniert nur, wenn die Programmbibliothek wie oben beschrieben zur Arduinosoftware hinzugefügt wurde). Das Programm ist für den UNO R3 Mikrocontroller vorgesehen. Bei MEGA2560 und anderen Controllern müssen die Pins entsprechend der zuvor in dieser Anleitung dargestellten Tabelle angepasst werden

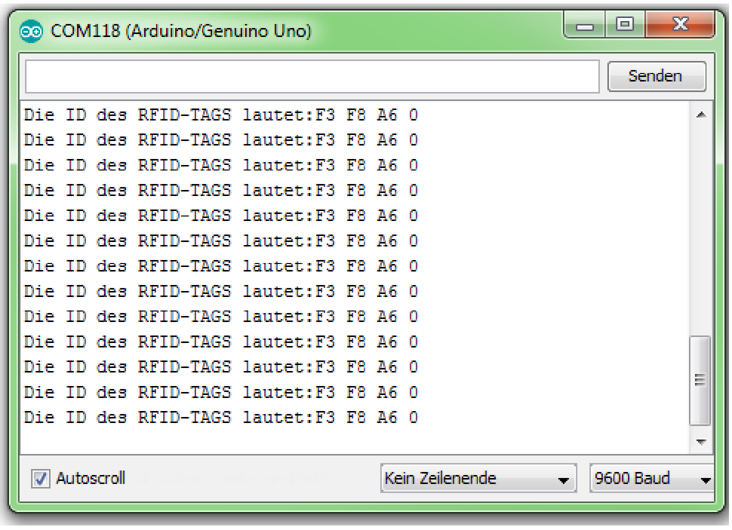
Sketch 1:

```
#include <SPI.h>                    //SPI-Bibiothek hinzufügen
#include <MFRC522.h>                //RFID-Bibiothek hinzufügen
#define SS_PIN 10                   //SDA an Pin 10 (bei MEGA anders)
#define RST_PIN 9                   //RST an Pin 9 (bei MEGA anders)
MFRC522 mfrc522(SS_PIN, RST_PIN);   //RFID-Empfänger benennen

void setup()                        //Beginn des Setups:
{
Serial.begin(9600);                 //Serielle Verbindung starten (Monitor)
SPI.begin();                        //SPI-Verbindung aufbauen
mfrc522.PCD_Init();                 //Initialisierung des RFID-Empfängers
}

void loop()                         //Hier beginnt der Loop-Teil
{
if ( ! mfrc522.PICC_IsNewCardPresent()) //Wenn eine Karte in Reichweite //ist...
{
return;                             //gehe weiter...
}
if ( ! mfrc522.PICC_ReadCardSerial()) //Wenn ein RFID-Sender ausgewählt wurde
{
return;                             //gehe weiter...
}
Serial.print("Die ID des RFID-TAGS lautet:");
//"Die ID des RFID-TAGS lautet:" wird auf den Serial Monitor geschrieben
for (byte i = 0; i < mfrc522.uid.size; i++)
{
Serial.print(mfrc522.uid.uidByte[i], HEX);
//Dann wird die UID ausgelesen,die aus vier einzelnen Blöcken besteht und der
Reihe nach an den Serial Monitor gesendet. Die Endung Hex bedeutet, dass die
vier Blöcke der UID als HEX-Zahl (also auch mit Buchstaben) ausgegeben wird
Serial.print(" ");
//Der Befehl „Serial.print(" ");“ sorgt dafür, dass zwischen den einzelnen
ausgelesenen Blöcken ein Leerzeichen steht.
}
Serial.println();
//Mit dieser Zeile wird auf dem Serial Monitor nur ein Zeilenumbruch gemacht.
}
```

Wenn alles funktioniert hat, sieht das Ergebnis am Serial-Monitor (Abgesehen von der eigenen UID) so aus:



Mit dieser hintereinander geschrieben HEX-Zahl lässt sich nicht gut arbeiten. Also ändern wir die Zeile „Serial.print(mfrc522.uid.uidByte[i], HEX);“ um in „Serial.print(mfrc522.uid.uidByte[i], DEC;“. Dann bekommt man als Ergebnis die einzelnen Blöcke des UID-Codes als Dezimalzahl ausgegeben.

RFID Sketch 2

Jetzt wird der UID-Code zwar als Dezimalzahl ausgegeben, aber er ist immernoch in vier Blöcke aufgeteilt. Wir verändern den Code jetzt mit ein wenig Mathematik dahingehend, dass wir für die UID eine einzige zusammenhängende normale Zahl erhalten (Dezimalzahl).

Warum machen wir das? Wenn wir später den Sketch verwenden wollen um etwas in Abhängigkeit eines richtig ausgelesenen RFID-TAGs auszulösen (Z.B. eine LED soll leuchten oder eine Servomotor soll sich um 90 Grad drehen…) können wir mit einer zusammenhängenden Zahl besser einen IF-Befehl verwenden. Zum beispiel:

„Wenn der RFID-Code=1031720 ist, dann soll eine LED für 5 Sekunden leuchten“.

Schwerer wäre es dagegen mit dem Befehl „Wenn der erste Block 195 lautet und der zweite Block 765 lautet und der dritte Block 770 lautet und der vierte Block 233 lautet … Dann schalte eine LED für 5 Sekunden an.

Nachteil ist jedoch, dass der Sketch dadurch etwas unsicherer wird, weil nicht alle Zahlen der vier Blöcke (Max. 12 Ziffern) in einer Zusammenhängenden Zahl dargestellt werden können. Wenn es sicherer sein soll, müsste man jeden einzelnen Block als solchen abfragen.

Sketch 2:

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup()
{
Serial.begin(9600);
SPI.begin();
mfrc522.PCD_Init();
}

void loop()
{
if ( ! mfrc522.PICC_IsNewCardPresent())
{
return;
}
if ( ! mfrc522.PICC_ReadCardSerial())
{
return;
}
long code=0;      //Als neue Variable fügen wir „code“ hinzu, unter welcher
                  später die UID als zusammenhängende Zahl ausgegeben wird.
                  Statt int benutzen wir jetzt den Zahlenbereich „long“, weil
                  sich dann eine größere Zahl speichern lässt.
for (byte i = 0; i < mfrc522.uid.size; i++)
{
code=((code+mfrc522.uid.uidByte[i])*10); //Nun werden wie auch vorher die vier
                                         Blöcke ausgelesen und in jedem
                                         Durchlauf wird der Code mit dem Faktor
                                         10 „gestreckt“. (Eigentlich müsste man
                                         hier den Wert 1000 verwenden, jedoch
                                         würde die Zahl dann zu groß werden.
}
Serial.print("Die Kartennummer lautet::"); //Zum Schluss wird der Zahlencode
                                            (Man kann ihn nicht mehr als UID
                                            bezeichnen) ausgegeben.

Serial.println(code);
}
```

Jetzt kann von einem RFID-TAG die eindeutige Identifikationsnummer ausgelesen werden. Sie wird wird am serial-monitor angezeigt. In unserem Fall lautet die Identifikationsnummer dieses individuellen RFID-TAGs „1031720“.

Und wie geht es weiter? Jetzt wollen wir eine LED für 5 Sekunden einschalten, wenn genau dieser RFID-TAG mit der Nummer „1031720“ vor den RFID-READER gehalten wird.

Sketch 3:

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup()
{
Serial.begin(9600);
SPI.begin();
mfrc522.PCD_Init();
pinMode (2, OUTPUT);    //Der Pin 2 ist jetzt ein Ausgang (Hierwird eine LED
                        angeschlossen)
}

void loop()
{
if ( ! mfrc522.PICC_IsNewCardPresent())
{
return;
}
if ( ! mfrc522.PICC_ReadCardSerial())
{
return;
}
long code=0;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
code=((code+mfrc522.uid.uidByte[i])*10);
}
Serial.print("Die Kartennummer lautet:");
Serial.println(code);

// Ab hier erfolgt die erweiterung des Programms.
if (code==1031720)                  //Wenn der Zahlencode 1031720 lautet...

{                                   //Programmabschniss öffnen
digitalWrite (2, HIGH);             //...dann soll die LED an Pin 2 leuchten...
delay (5000);                       //für 5 Sekunden
digitalWrite (2, LOW);              // … und danach wieder aus gehen
}                                   //Programmabschnitt schließen
}                                   //Sketch abschließen
```